Program Analysis – Lecture 17

Call Graph Analysis

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart
Winter 2019/2020
What does this Java code print?

class Reflection {
    static class Car {
        private String color;
        protected void getColor() {
            System.out.println("A "+color+" car");
        }
    }

    public static void main(String[] args)
        throws Exception {
        Class clazz = Class.forName("Reflection$Car");
        Car car = (Car) clazz.newInstance();
        Method getColor = clazz.getDeclaredMethod("getColor");
        getColor.invoke(car);
    }
}

https://ilias3.uni-stuttgart.de/vote/KN2I
Warm-up Quiz

What does this Java code print?

class Reflection {
    static class Car {
        private String color;
        protected void getColor() {
            System.out.println("A "+color+" car");
        }
    }
    public static void main(String[] args) throws Exception {
        Class clazz = Class.forName("Reflection$Car");
        Car car = (Car) clazz.newInstance();
        Method getColor = clazz.getDeclaredMethod("getColor");
        getColor.invoke(car);
    }
}

Result: A null car

https://ilias3.uni-stuttgart.de/vote/KN2I
Call Graph Analysis

- **Call graph**: Abstraction of all method calls in a program
  - Nodes: Methods
  - Edges: Calls
  - Flow-insensitive: No execution order

- **Here**: Static call graph
  - Abstraction of all calls that *may* execute
See Ilias

Remaining slides (courtesy of Eric Bodden):
Stored in Ilias